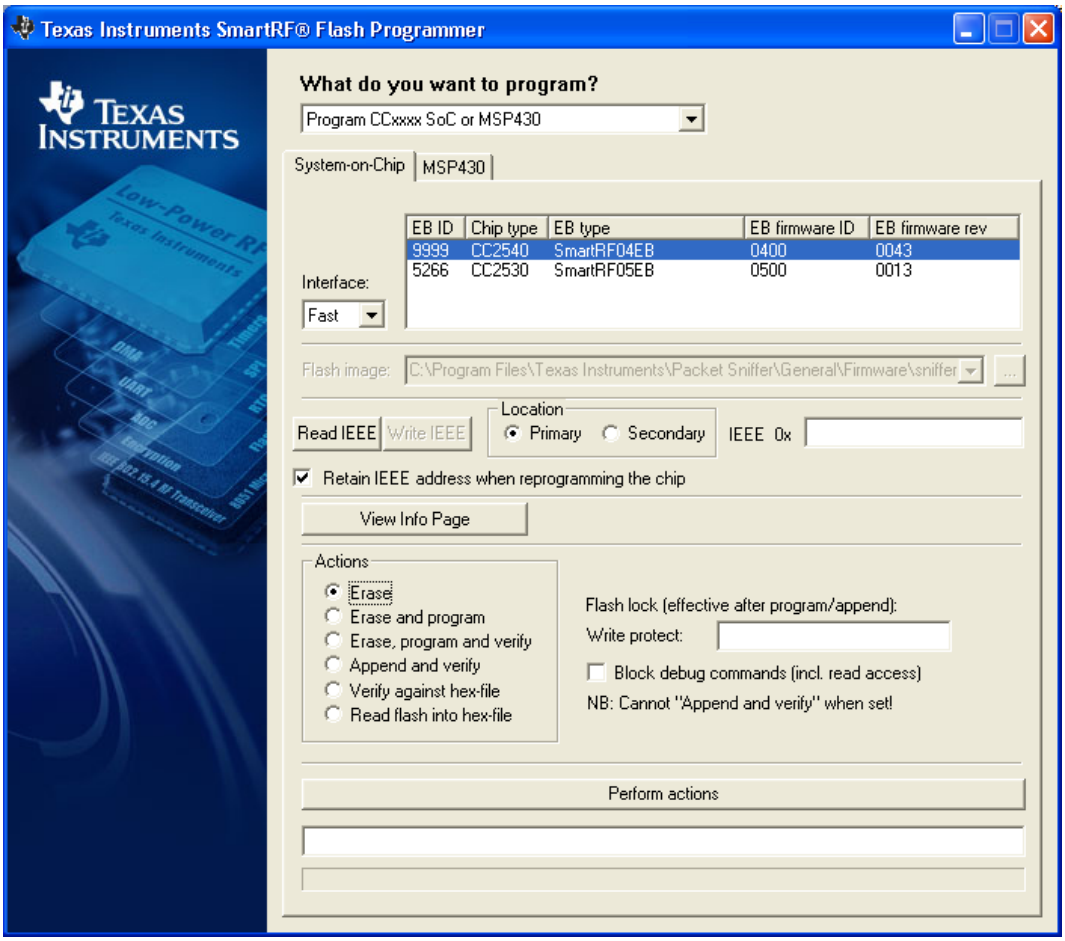
SmartRF™ Flash Programmer User Manual



SWRU069E

Table of contents

1 INTRODUCTION 3
2 ABOUT THIS MANUAL 3
3 DEFINITIONS 3
4 PROGRAMMING USING THE GUI VERSION 4
4.1 PROGRAM CCXXX SOC OR MSP430 DEVICES 4
4.1.1 System on chip 5
4.1.2 MSP430 Programming 10
4.2 PROGRAM EVALUATION BOARD 11
4.2.1 USB MCU firmware update 11
4.2.2 Automatic Firmware update of the Evaluation Board. 12
4.2.3 EB bootloader 14
5 COMMAND LINE INTERFACE 16
5.1 OPTIONS 16
5.2 PLUG-IN TO IAR WORKBENCH 16
5.2.1 Setup 16
5.2.2 Use 17
6 INSTALLED HEX FILES 20
7 DOCUMENT HISTORY 20

1 Introduction

This is the user manual for the SmartRF Flash Programmer.

The Flash Programmer can be used to program the flash memory in Texas Instruments Low Power RF System on Chips and for programming the flash memory of MSP430 devices via the MSP-FET430UIF and the eZ430 dongle.

For IEEE 802.15.4 compliant devices (e.g. CC2530) and Bluetooth® low energy devices (e.g. CC2540) the Flash Programmer support reading and writing the IEEE/MAC address.

In addition, the Flash Programmer can be used for upgrading the firmware on the SmartRF04EB, SmartRF05EB, CC Debugger and CC2430DB.

2 About this manual

This manual covers the use of the Flash programmer, both the GUI version and the -Command Line Interface.

The manual describes the most common functions and options available. Chapter 5.2 will describe how to use the Command Line Interface of the Flash Programmer from IAR Embedded Workbench to perform post-build operations like flash programming.

The Flash Programmer has functionality to program the USB MCU found on SmartRF04EB and CC2430DB through the Silicon Laboratories serial adapter EC2. This is not covered in this manual.

3 Definitions

CC Debugger	The CC Debugger can be used as interface to program SoC's mounted on Battery Boards and to update the USB MCU on SmartRF05EB.
CLI	Command Line Interface
Factory firmware	The firmware that is supplied programmed into the USB MCU from the factory. This firmware supports SmartRF® Studio operation as well as a stand-alone PER tester.
GUI	Graphical User Interface
SmartRF®04DK	A collective term used for all development kits for the SmartRF®04 platform, i.e. CC2510DK and CC2430ZDK
SmartRF®05DK	A collective term used for all development kits for the SmartRF05 platform, i.e. CC2520DK
USB MCU	The Silicon Labs C8051F320 MCU used to provide a USB interface on the SmartRF®04EB and CC2430DB. The CC2511 MCU used to provide a USB interface on the SmartRF05EB.

4 Programming Using the GUI Version

The Graphical User Interface operates in two different user modes: One for programming of System on Chip modules and one for programming of the Evaluation Boards MCU.

4.1 Program CCxxx SoC or MSP430 devices.

Figure 1 shows the user mode for programming of System on Chip modules. Two types of modules are supported.

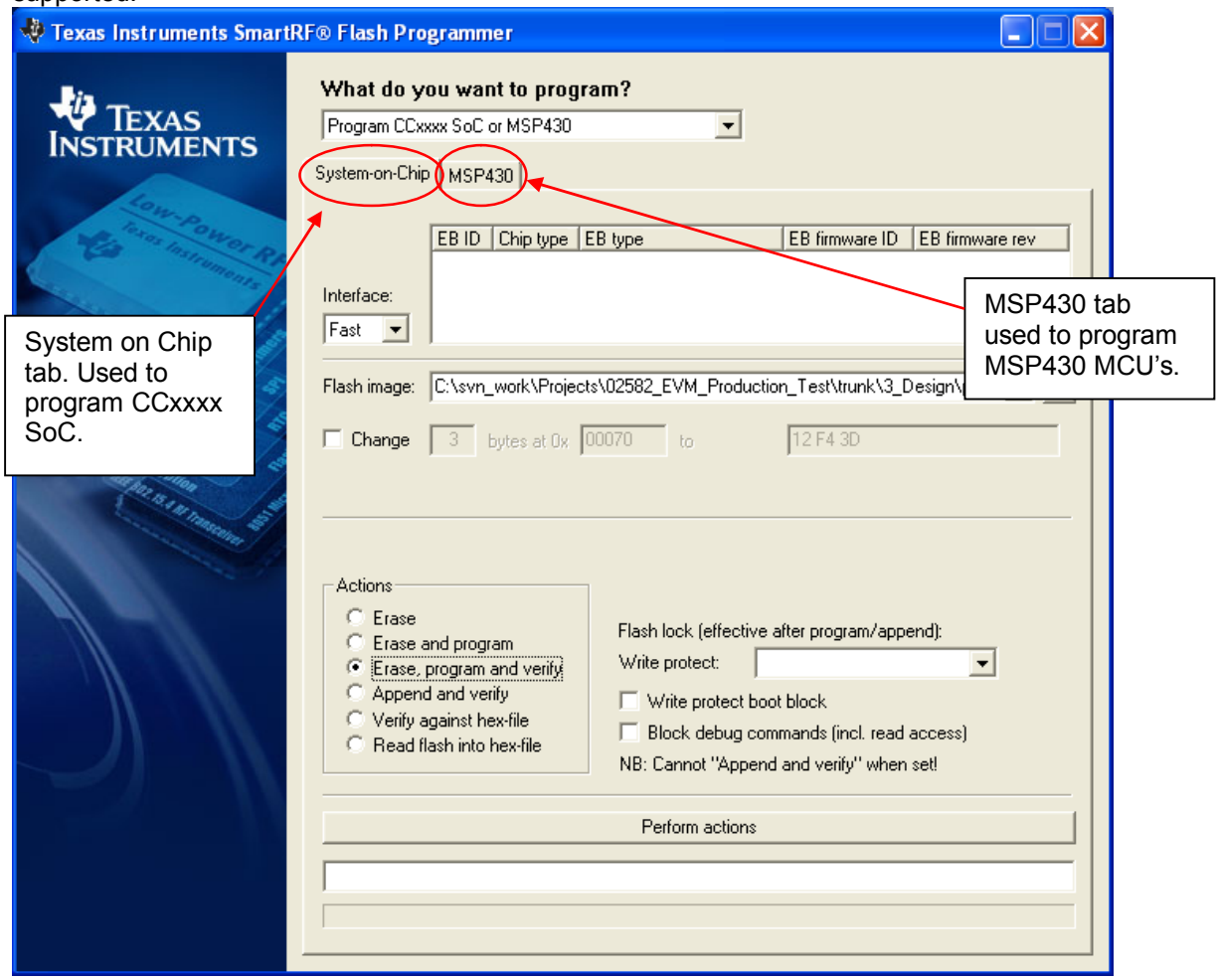


Figure 1, Program CCxxxx SoC or MSP430

"System on Chip" is used to program Texas Instruments SOC's e.g. CC1110, CC2430, CC2510, and CC2530. The use of this tab is described in chapter 4.1.1.

"MSP430" is used to program the MSP430 MCU used in various RF development kits. Further details are described in chapter 4.1.2.

4.1.1 System on chip

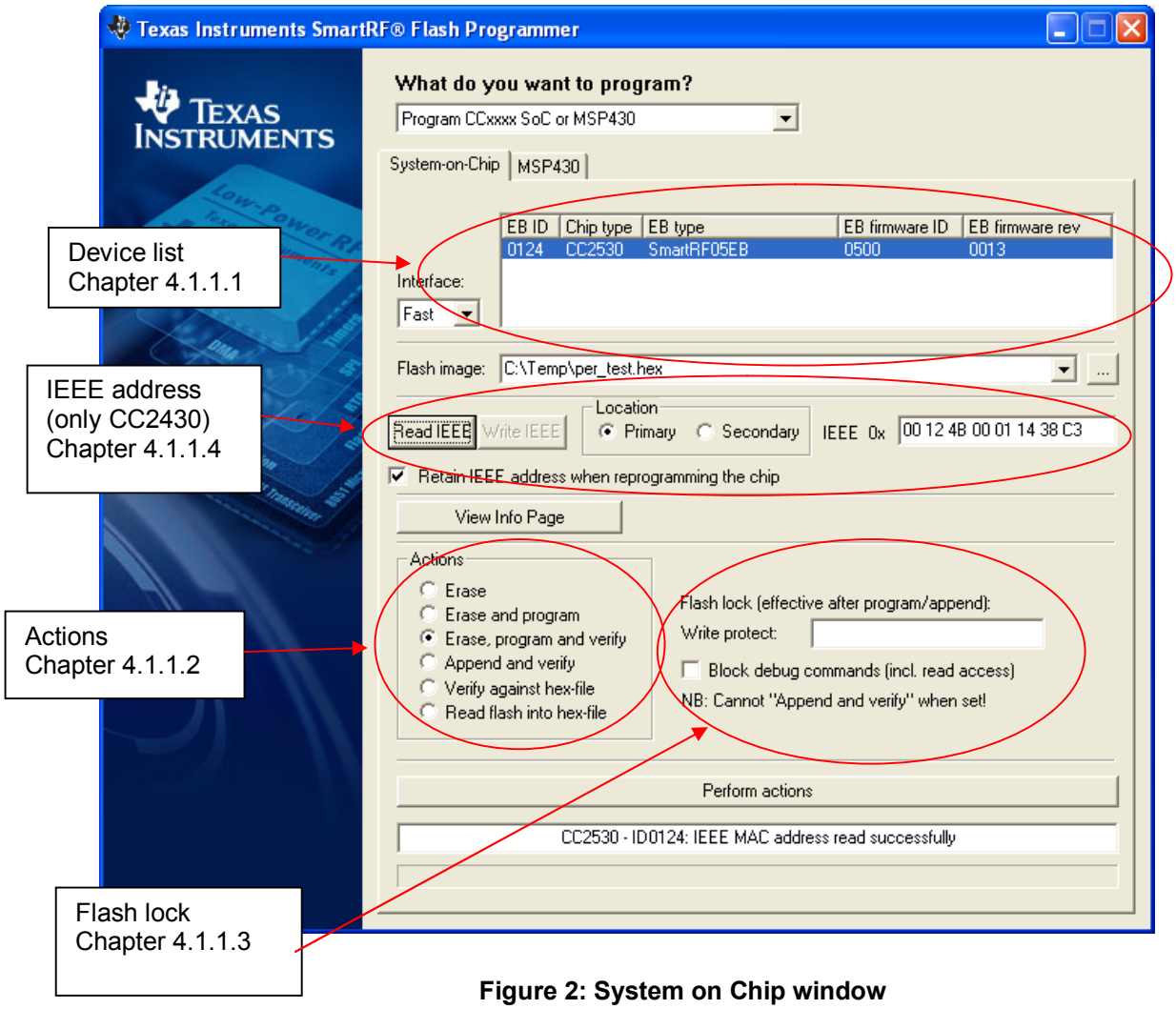


Figure 2: System on Chip window

4.1.1.1 Device list:

The device list show all currently connected System on Chip devices. Note that when the System on Chip tab is selected, any Evaluation Board without a System on Chip EM connected will not be displayed.

If more than one chip is connected the one selected (marked blue) in this window is the one that will be programmed.

4.1.1.2 Actions:

There are six different actions that can be performed on the Texas Instruments SoC. To perform an action, select one and then press the “Perform actions” button.
The progress bar and output window at the bottom will output the progress and result of the action.

The six actions are:

Erase
This action will erase the flash memory of the selected SoC.

Erase and program
Will erase the flash memory of the selected SoC and then program it with the .hex file selected in the **“**Flash image” field.

Erase, program and verify
Same as “Erase and program”, but after the programming the content of the flash will be read back and compared with the .hex file. This will detect errors during programming or errors caused by damaged flash. It is therefore recommended to always verify after programming.

Append and verify
This action will write the contents of the hex file given in the **“**Flash image” field, to the selected SoC without erasing the Flash first. Note that all the Flash written to must read 0xFF (be erased) before programming starts. Feature is useful when a program is divided into more that one hex file.
This action uses debug commands to read from the Flash, which means that if the debug commands are blocked on the chip, it is impossible to perform this action.

Verify against hex-file
This action will compare the contents of the Flash with a .hex file given in the **“**Flash image” field. Note that the function only verifies that the contents of the .hex file is present in the Flash, it does not check if there is anything additional written in the Flash.
This action uses debug commands to read from the Flash, which means that if the debug commands are blocked on the chip, it is impossible to perform this action.

Read into hex-file
This action will read the entire content of the Flash and then write it to the hex-file given in the **“**Flash image” field**.**
Note that the hex-file given in the **“**Flash image” field will be overwritten.
This action uses debug commands to read from the Flash, which means that if the debug commands are blocked on the chip, it is impossible to perform this action.

4.1.1.3 Flash lock:

When programming a chip it is possible to apply the different flash lock and debug command lock options that are supported by the chip. These fields will change depending on the chip type selected in the device list. Please refer to the datasheet for the different chip types for a description of these lock features.

Note that if the debug command lock is set, it is impossible to use most of the debug commands on the chip. E.g. the flash may no longer be read out.

CC11xx, CC25xx and CC24xx:

For these devices it is possible to write protect all pages or the upper part of the page numbers.

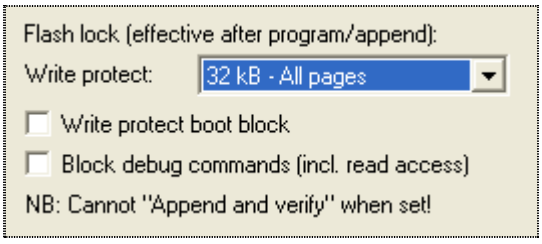

Flash lock (effective after program/append):
Write protect:
32 kB - All pages
Write protect boot block
Block debug commands (incl. read access)
NB: Cannot "Append and verify" when set!

Figure 3, Write protect CC243x

CC253x and CC2540:

For these devices it is possible to write protect each page of the flash image. The input field “Write protect” should be given as shown in Figure 4, Write protect CC2530. The pages to be write protected can be given separated by a comma. It is also possible to specify a range of pages.

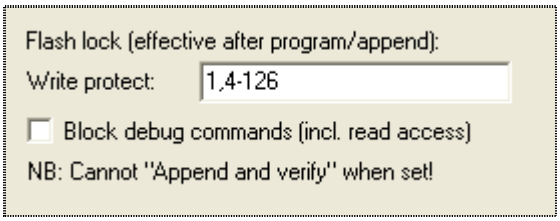

Flash lock (effective after program/append):
Write protect:
1,4-126
Block debug commands (incl. read access)
NB: Cannot "Append and verify" when set!

Figure 4, Write protect CC2530

4.1.1.4 IEEE 802.15.4 address / general change field:

The input fields for the IEEE address depend on the connected RF Device.

Figure 5: IEEE address for ZigBee SoC (CC2430/31)

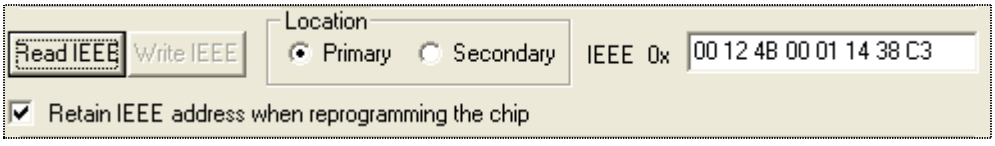


Figure 6: IEEE address for ZigBee SoC (CC2530/31)

Figure 7: Change field for non ZigBee SoC

IEEE address on ZigBee devices like CC2430/31

On a CC243x the IEEE address is stored in the last 8 bytes of the flash. E.g. the placement is different depending on the size of the Flash. See Table 1 below.

Chip type	IEEE address start	IEEE address end
CC243xF128	0x1FFF8	0x1FFFF
CC243xF64	0xFFF8	0xFFFF
CC243xF32	0x7FF8	0x7FFF

Table 1: Placement of IEEE address

To read the IEEE address from a chip select the appropriate Chip type (e.g. F-128) and push the "Read IEEE" button.

To write the IEEE address to a chip, manually write the address into the IEEE field (hexadecimal, with a space between each byte) and then push the "Write IEEE address" button.

Writing the IEEE address will fail if the flash is write-protected, or the debug command lock is set.

If the "Retain IEEE address when reprogramming the chip" is checked the IEEE address is preserved when a new program is written to the chip with the "Erase and program" or "Erase, program and verify" action. This is however not possible if the debug command lock is set on the chip before the programming starts.

IEEE address on ZigBee devices like CC2530/31

For CC253x it is possible to have two IEEE addresses programmed in flash. The Primary address is programmed in the Information Page and can only be read. The address is preprogrammed in factory.

The secondary IEEE address, which is optional (used when the address in the information page is not used), is stored at the end of the flash. The last 16 bytes is used for lock bits and the IEEE address is stored in the last 8 bytes before the lock bits.

Chip type	IEEE address start	IEEE address end
CC253xF256	0x3FFE8	0x3FFEF
CC253xF128	0x1FFE8	0x1FFEF
CC253xF64	0xFFE8	0xFFEF
CC253xF32	0x7FE8	0x7FEF

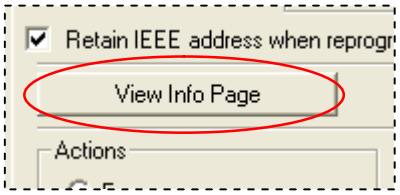
To read the IEEE address, select either Primary or Secondary and push the "Read IEEE" button.

Only the secondary IEEE address can be written. All the other rules are the same as described above for CC243x.

Change Field on non ZigBee devices

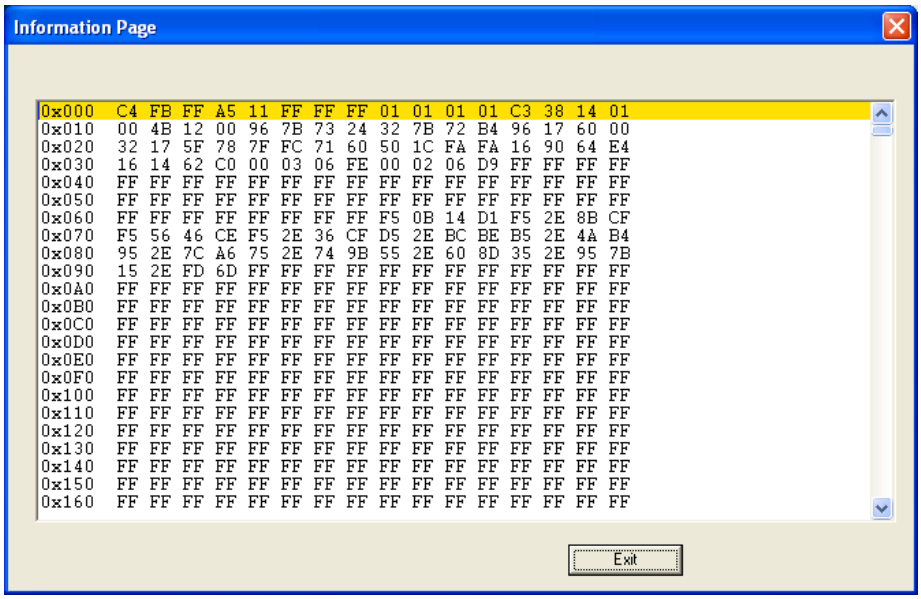
The intention of this field is to provide an easy and quick way to give a unique address to the chip when programming it. It gives the user the possibility to change any number of bytes at any location in the program read from the hex file, before it is written to the chip.
When "Change" is checked, input the start address, e.g. the first byte that should be changed into the first field.
Then the new values are written into the rightmost field (hexadecimal, with a space between each byte)
When "Erase and program" or "Erase, program and verify" action is performed, the bytes at the given address from the hex file are replaced with those written by the user before the chip is programmed. The hex file itself is not changed.

4.1.1.5 View Information Page



To view the information page click on the “View Info Page” button shown at the left side (Only applicable for CC253x and CC254x).

The **Information Page** is a read-only region that stores various device information.


Information Page
0x000 C4 FB FF A5 11 FF FF FF 01 01 01 01 C3 38 14 01
0x010 00 4B 12 00 96 7B 73 24 32 7B 72 B4 96 17 60 00
0x020 32 17 5F 78 7F FC 71 60 50 1C FA FA 16 90 64 E4
0x030 16 14 62 C0 00 03 06 FE 00 02 06 D9 FF FF FF FF
0x040 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x050 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x060 FF FF FF FF FF FF FF FF F5 0B 14 D1 F5 2E 8B CF
0x070 F5 56 46 CE F5 2E 36 CF D5 2E BC BE B5 2E 4A B4
0x080 95 2E 7C A6 75 2E 74 9B 55 2E 60 8D 35 2E 95 7B
0x090 15 2E FD 6D FF FF FF FF FF FF FF FF FF FF FF FF
0x0A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x0B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x0C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x0D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x0E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x0F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x100 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x110 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x120 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x130 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x140 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x150 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x160 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
Exit

4.1.2 MSP430 Programming

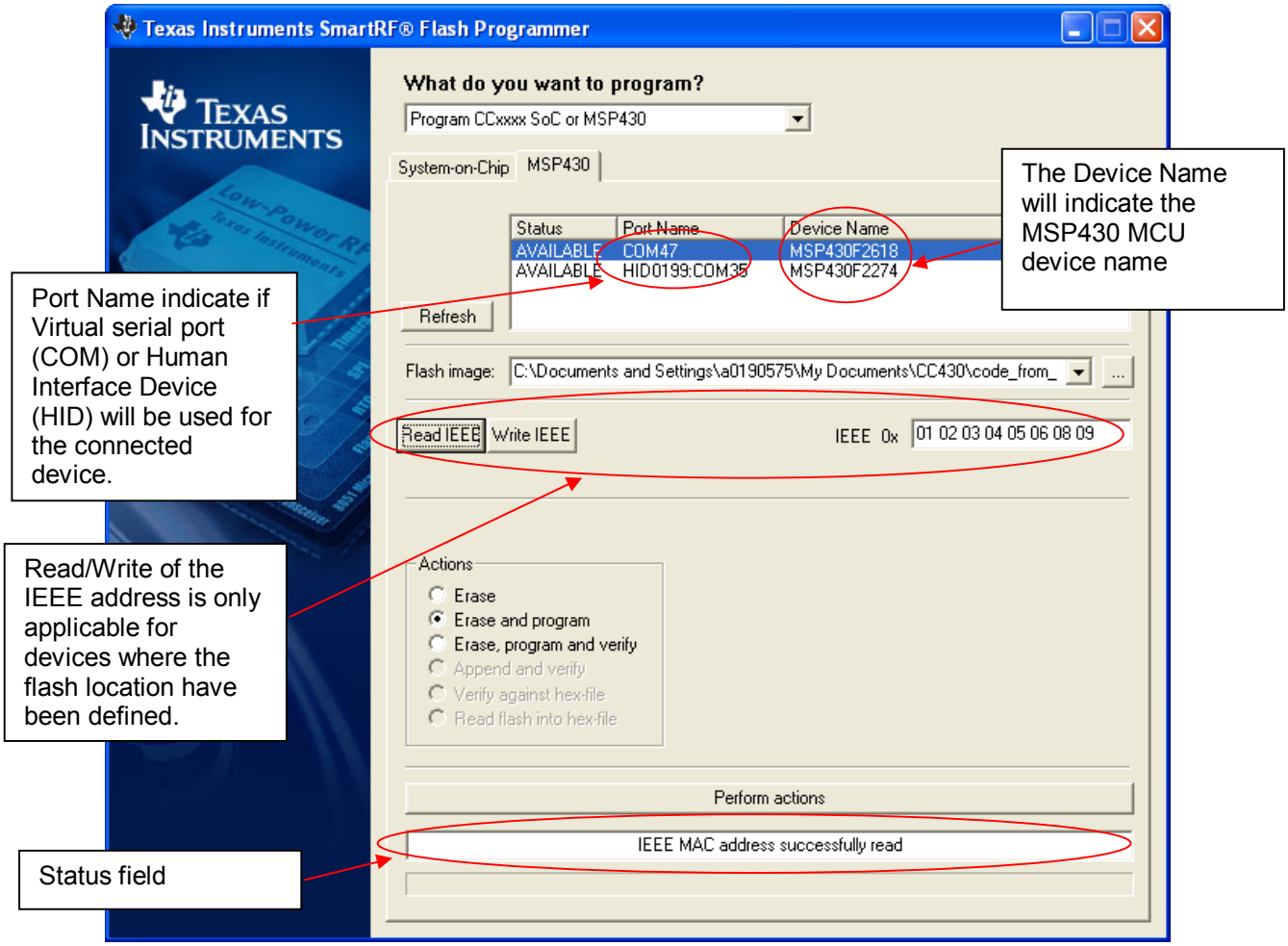


Figure 8: MSP430 Programming.

Figure 8 show the tab used for programming of the MSP430 MCU. The device can be connected via the USB-Debug-Interface (MSP-FET430UIF) or via the eZ430 USB dongle. The figure show both cases. Devices connected with the MSP-FET430UIF will appear as a COM port. In this case it is COM47. For eZ430 connected devices it will be seen as both a HID port and COM port.

When the device is connected via the USB interface, it could take a few seconds before the device appear in the device list.

The status of all actions will be given in the Status field at the bottom of the window. An attempt to program a hex file build for another MCU family will be detected and reported in the Status field.

The Firmware version of the MSP-FET430UIF will be checked automatically when a device is connected. If the FW version does not match the PC Software version, a message will be given and the user must choose whether or not to update the FW. The update will be performed automatically if the user chooses to update the FW.

4.2 Program Evaluation Board

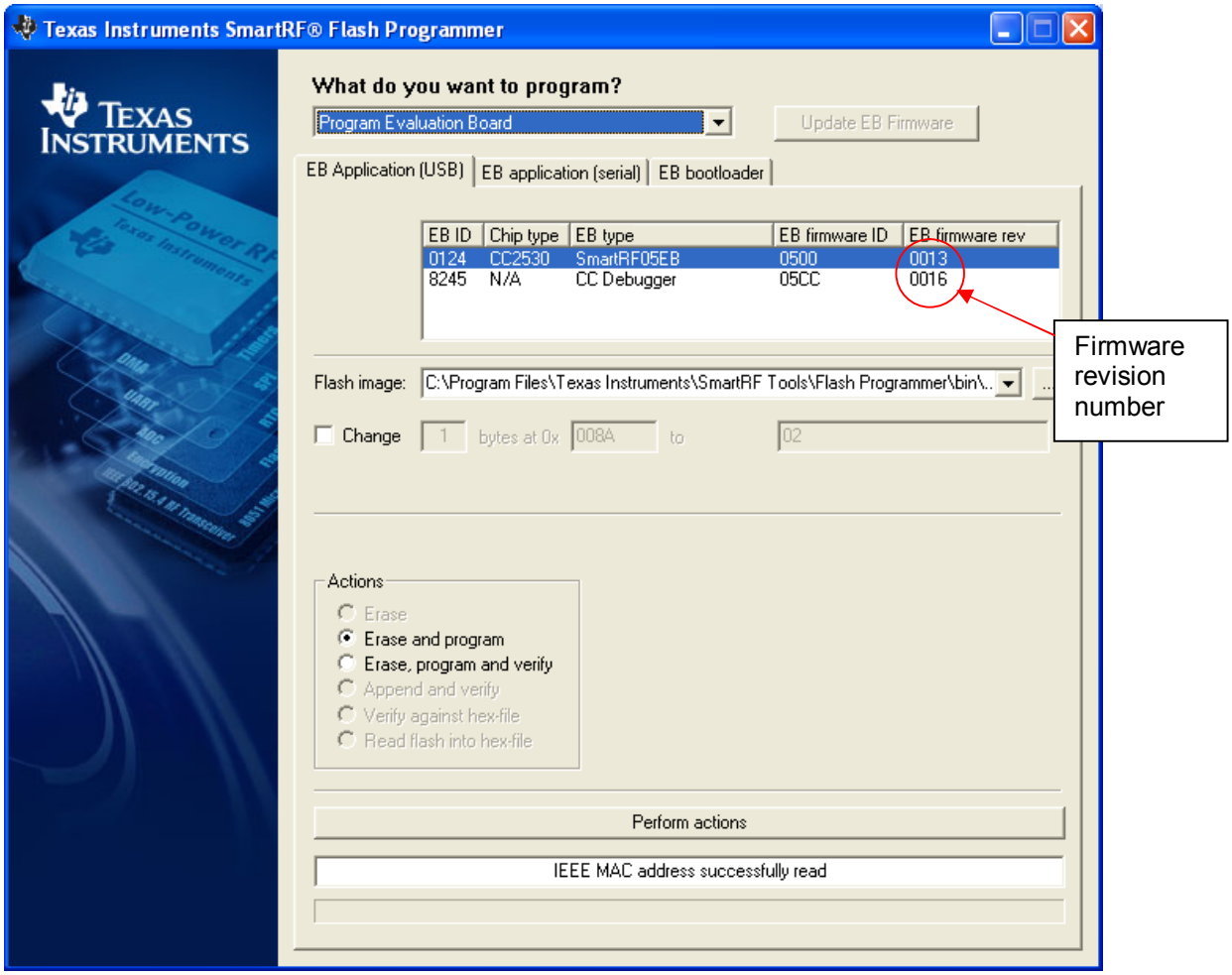


Figure 9, Programming of the Evaluation Board

“EB application (USB)” is used when updating the USB MCU found on SmartRF04EB, SmartRF05EB, CC Debugger and CC2430DB. The use of this tab is described in chapter 4.2.1.

“EB application (serial)” can be used in stead of the “EB application (USB)” to update the USB MCU on SmartRF04EB. This setup requires the EC2 serial adapter from Silab. For the rest it has the same functionality as the “EB application (USB)”.

“EB bootloader” is used to update the bootloader on SmartRF04EB, SmartRF05EB, CC Debugger and CC2430DB. Further details are described in chapter 4.2.3.

4.2.1 USB MCU firmware update

Figure 9 show the “EB application (USB)” tab. It provides the possibility to update the firmware on an Evaluation Board using only a USB cable. No additional programmer is necessary. When a SmartRF04EB, SmartRF05EB, CC Debugger or CC2430DB is connected it will appear in the device list. In the rightmost column the revision number of the current firmware can be read.
Note that the update procedure is different for SmartRF04EB and CC2430DB. However the hex file used, (fw400.hex), is identical for the two products.

SmartRF05EB and CC Debugger uses a different hex file specially build for the USB MCU (CC2511)

4.2.1.1 *Updating SmartRF04EB/SmartRF05EB USB MCU firmware*

1. Remove any CCxxxxEM module and all external equipment connected to the Evaluation Board.
2. Connect the USB cable to the Evaluation Board and turn it on, it should appear in the Device list with “Chip type” N/A.
3. Browse to the correct flash image (e.g. fw0400.hex for SmartRF04EB)
4. Choose the “Erase, program and verify”
5. Push “Perform actions”.
6. The status indicator at the bottom will show the progress and when completed the text “EB firmware update OK” will appear.

4.2.1.2 *Updating CC2430DB USB MCU firmware*

1. Remove all jumpers on P5.
2. Connect pin 9 and 10 on P4 (USB deb) together.
3. Connect the USB cable to the CC2430DB and turn it on, it should appear in the Device list with “Chip type”, “EB type” and “EB firmware ID” set to N/A.
4. Browse to the correct flash image (e.g. fw400.hex)
5. Choose the “Erase, program and verify”
6. Push “Perform actions”.
7. The status indicator at the bottom will show the progress and when completed the text “EB firmware update OK” will appear.
8. Remove jumper on pin 9-10 on P4, and mount jumpers on P5.

Note: After the programming is finished it takes a few seconds before the device appear in the device list. This is due to timing constrains on the USB bus after programming and reset of the device.

4.2.2 *Automatic Firmware update of the Evaluation Board.*

The firmware can be updated automatically with the latest version. The latest version of the firmware is installed together with any of the SmartRF Tools. When the Evaluation Board is connected to the USB port, the flash programmer will compare the current firmware version with the version found in the hex files installed together with the flash programmer. If current firmware is found to be an old version, this will be indicated as shown in Figure 10.

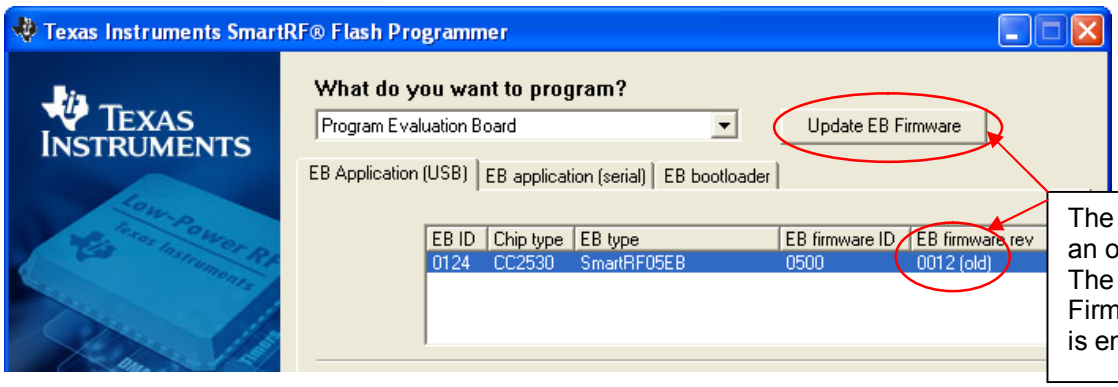


Figure 10, Old firmware indication

There will also be a popup dialog with information about how to update the EB Firmware. See Figure 11

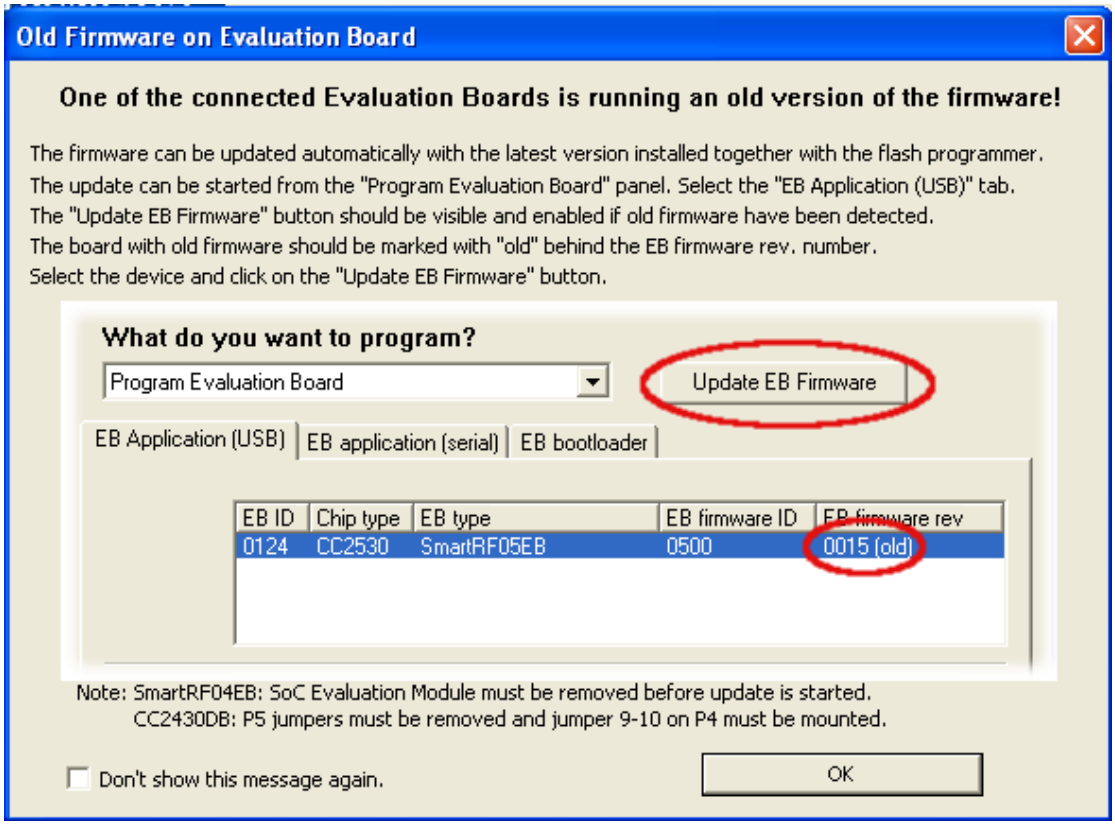

Old Firmware on Evaluation Board
One of the connected Evaluation Boards is running an old version of the firmware!
The firmware can be updated automatically with the latest version installed together with the flash programmer.
The update can be started from the "Program Evaluation Board" panel. Select the "EB Application (USB)" tab.
The "Update EB Firmware" button should be visible and enabled if old firmware have been detected.
The board with old firmware should be marked with "old" behind the EB firmware rev. number.
Select the device and click on the "Update EB Firmware" button.
What do you want to program?
Program Evaluation Board
Update EB Firmware
EB Application (USB)
EB application (serial)
EB bootloader
EB ID
Chip type
EB type
EB firmware ID
EB firmware rev
0124
CC2530
SmartRF05EB
0500
0015 (old)
Note: SmartRF04EB: SoC Evaluation Module must be removed before update is started.
CC2430DB: P5 jumpers must be removed and jumper 9-10 on P4 must be mounted.
Don't show this message again.
OK

Figure 11, Popup dialog for old EB firmware.

4.2.3 EB bootloader

The bootloader of the different Evaluation Boards can be updated from the “EB bootloader” tab.

4.2.3.1 SmartRF04EB

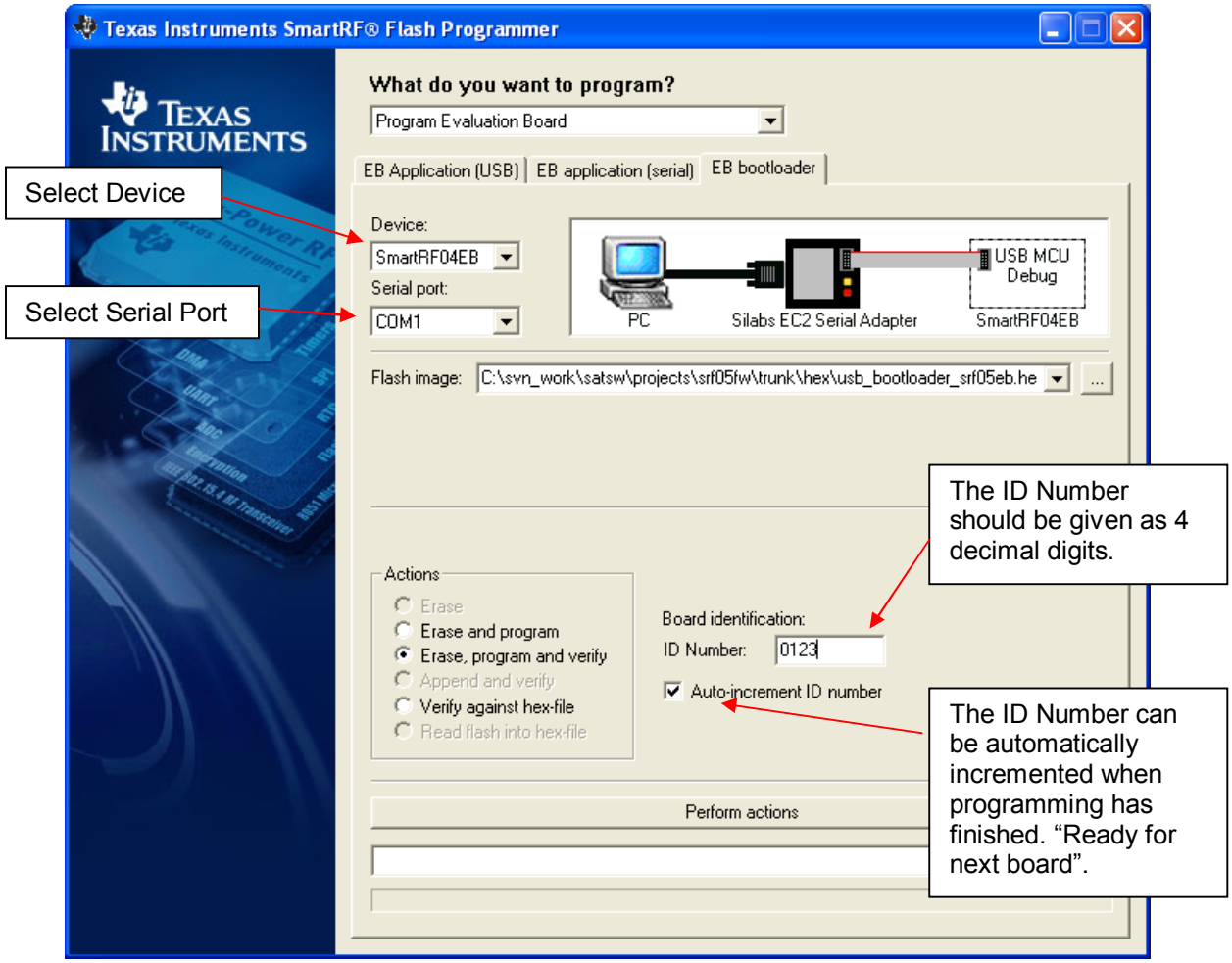


Figure 12, SmartRF04EB Bootloader

To program the bootloader on SmartRF04EB it is required to use the Serial Adapter (EC2) from Silabs.

Select Serial port and flash image. The flash image can be selected from a drop down list with a history of the last 10 images that has been programmed. It is also possible to use the button on the right side to browse for the required flash image

Specify the board identification (ID Number) and select the actions “Erase and program” or “Erase, program and verify”.

Click the “Perform actions” button to start programming. Status will be displayed in the field below the button.

The action “Verify against hex-file” can be used to check current image on the USB MCU with the given hex file.

4.2.3.2 SmartRF05EB

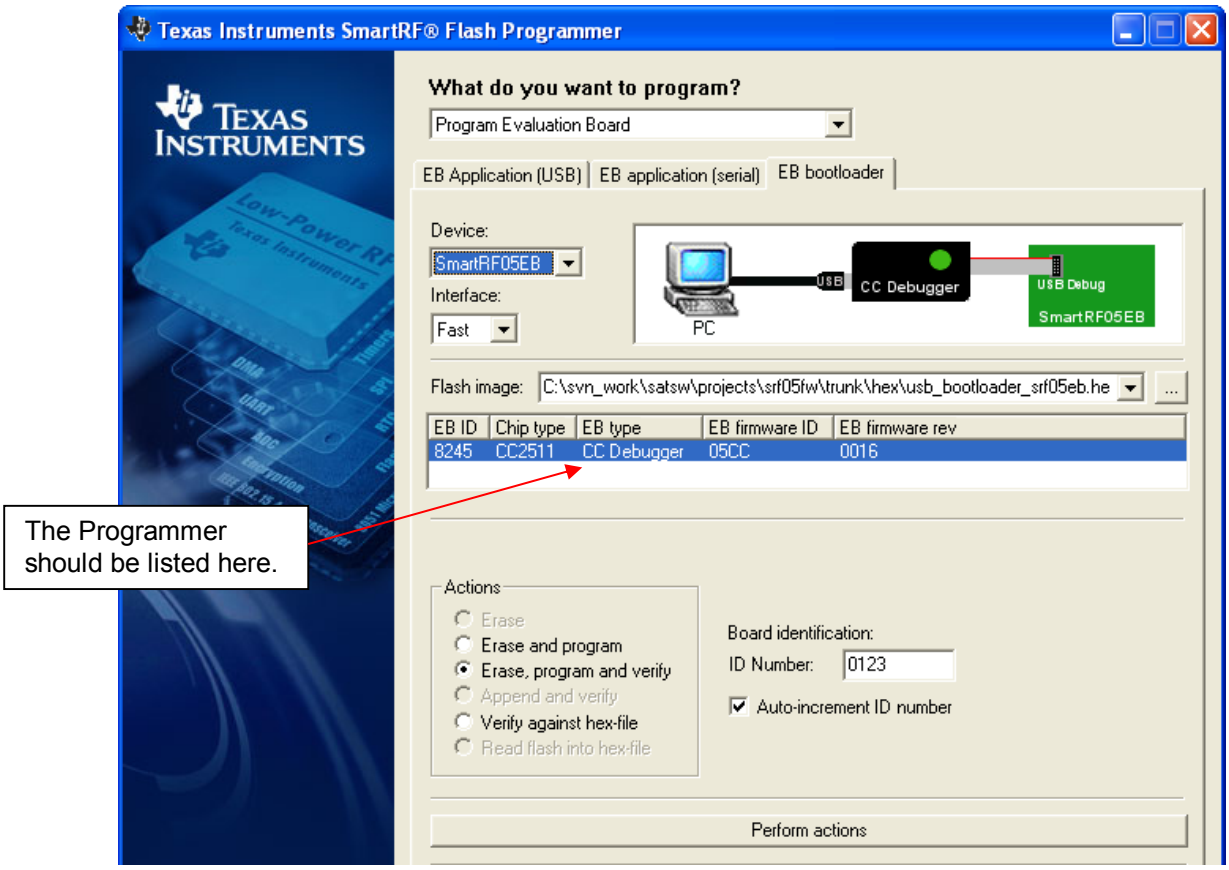


Figure 13, SmartRF05EB bootloader

For programming of the bootloader on SmartRF05EB, it is possible to use the CC Debugger as illustrated on the image shown when Device SmartRF05EB is selected. It is also possible to use a SmartRF04EB or SmartRF05EB board instead of the CC Debugger. The 10 pin flat ribbon cable should then be connected on the “SoC Debug” header on the SmartRF0xEB.

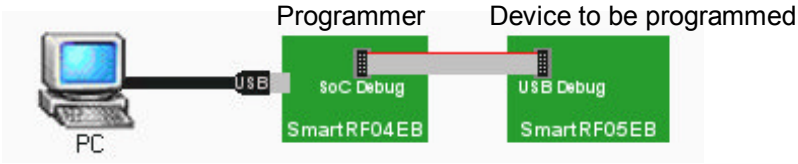


Figure 14, Programming of bootloader with SmartRF04EB

The interface speed can be set to either Fast or Slow. This determines the clock speed on the debug interface of the USB MCU. Normally there shouldn’t be any problem to use the fast speed.

The flash image can be selected from a drop down list with a history of the last 10 images that has been programmed. It is also possible to use the button on the right side to browse for the required flash image.

The connected CC Debugger or SmartRF04EB board should be visible in the list of connected devices. The “Chip type” should be CC2511. Select the required device.

Specify the board identification (ID Number) and select the actions “Erase and program” or “Erase, program and verify”.

Click the “Perform actions” button to start programming. Status will be displayed in the field below the button.

5 Command Line Interface

5.1 Options

To get all available options in the command line interface, run the SmartRFProgConsole.exe in a command window without any parameters/arguments. A list of all available options will then be printed out. These options are the same as the ones available in the GUI version of the Flash programmer. Please refer to chapter 4 for a description of these.

5.2 Plug-in to IAR Workbench

The command line interface can be integrated in the IAR Workbench. To setup IAR with this feature follow the instructions below.

5.2.1 Setup

Start IAR Workbench and choose “Configure Tools…”, from the Tools menu, Figure 15.

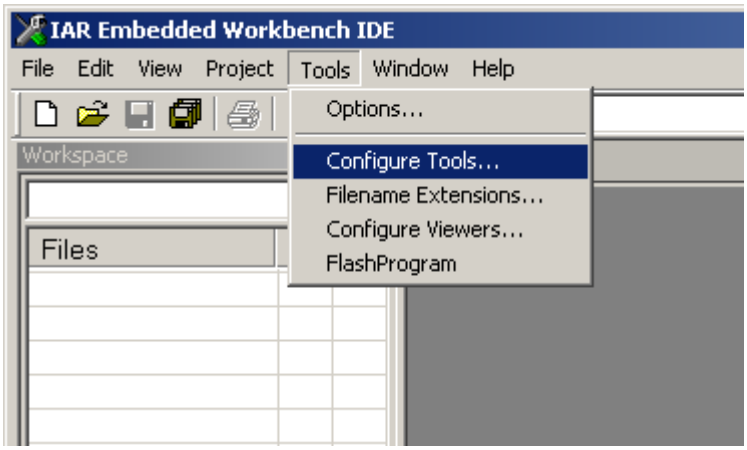


Figure 15: Tools Menu

Press “New”, and add the information present in Table 2, see Figure 16.

Field	Value
Menu Text:	FlashProgram
Command:	C:\Program Files\Texas Instruments\SmartRF Tools\Flash Programmer\bin\SmartRFProgConsole.exe[1]
Argument:	S() EPV F=$TARGET_PATH$ K(0)

Table 2: Flash Programmer Setup

[1] Insert the complete path to the Command Line Flash Programmer

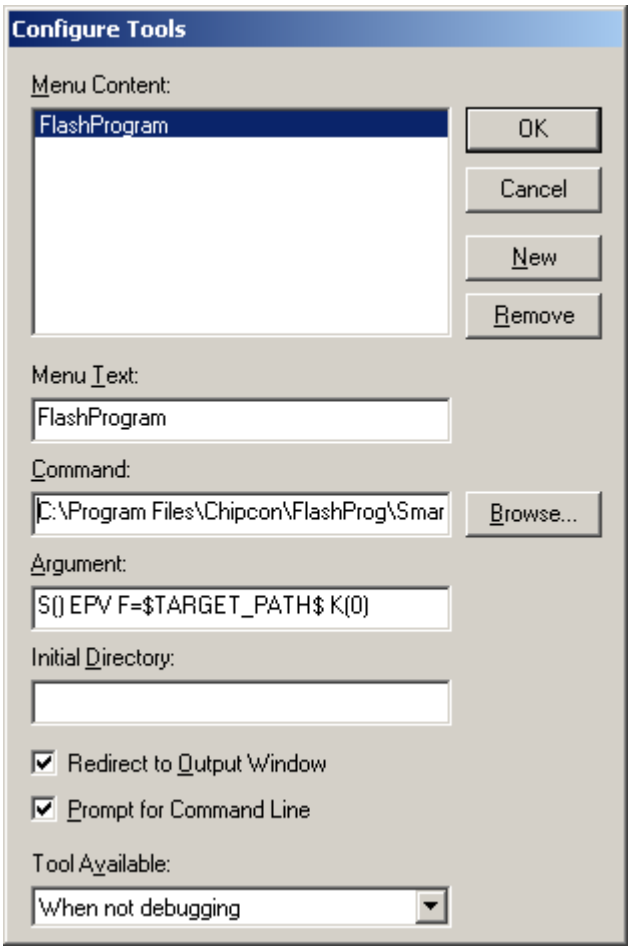


Figure 16: Configure Tools

5.2.2 Use

After setup, a new target is placed on the Tools menu.

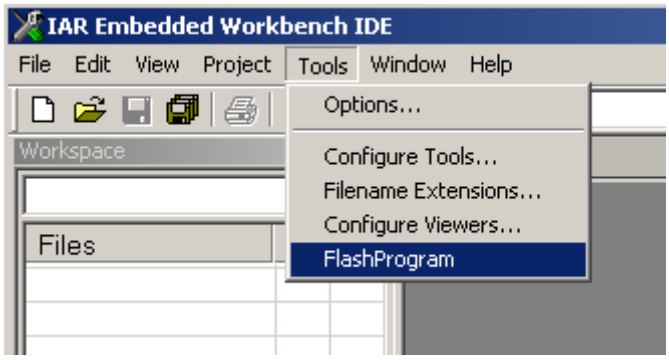


Figure 17: Using Flash Programmer from IAR Workbench

Setup your project to generate hex file as primary output (Figure 19), compile and link, and choose "Flash Program" from the Tools menu. A command line window will be displayed, Figure 18. After the "S" option an empty parenthesize is present. If this parenthesize is empty, the first available development card is used. If more than one development card is connected, fill in the ID number for the card you want to use in the empty parenthesize.

The K(0) option will retain the IEEE address while programming.
Use K(0) on CC2430F128, K(1) on CC2430F64 and K(2) on CC2430F32.
If the K option is removed the IEEE address is not retained.

The "EPV" option is for "Erase, program and verify"

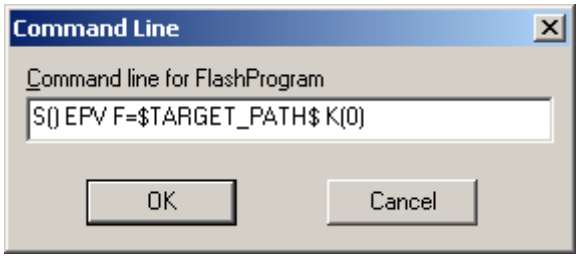


Figure 18: Command Line Window

Press OK and the hex file will be downloaded.

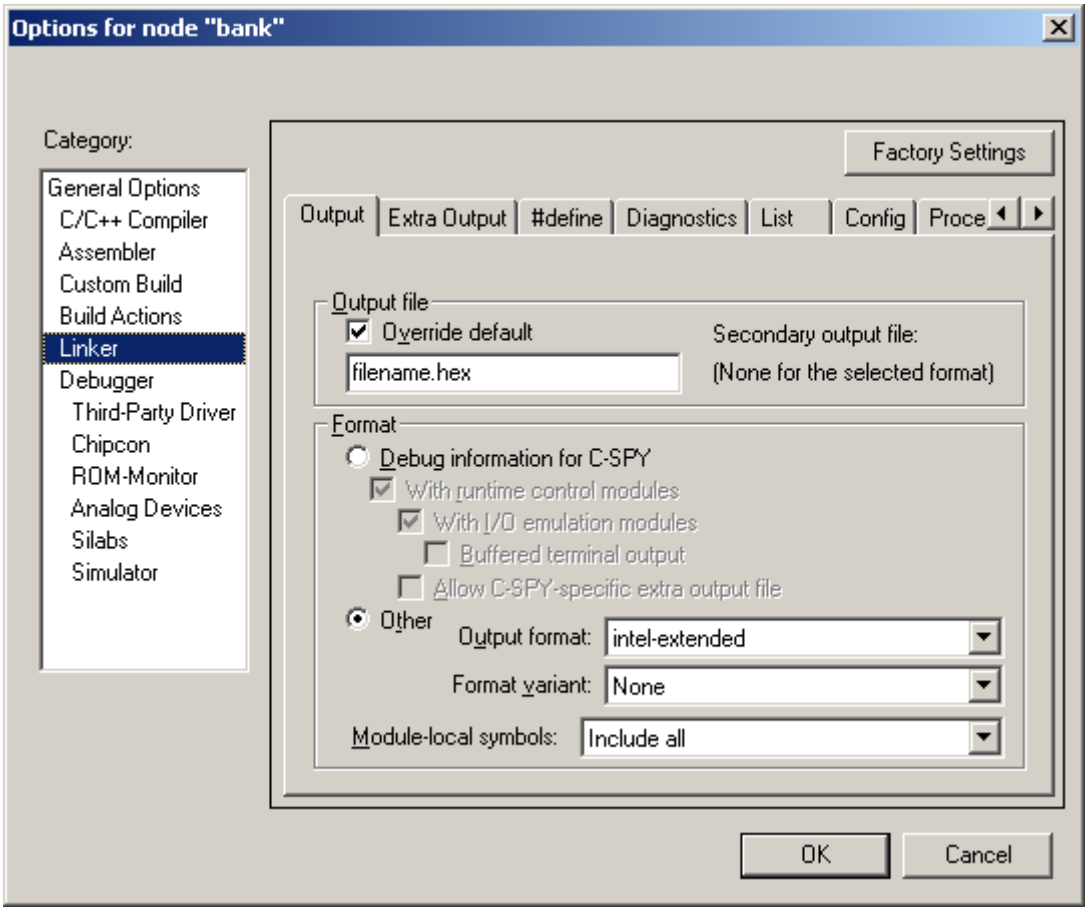
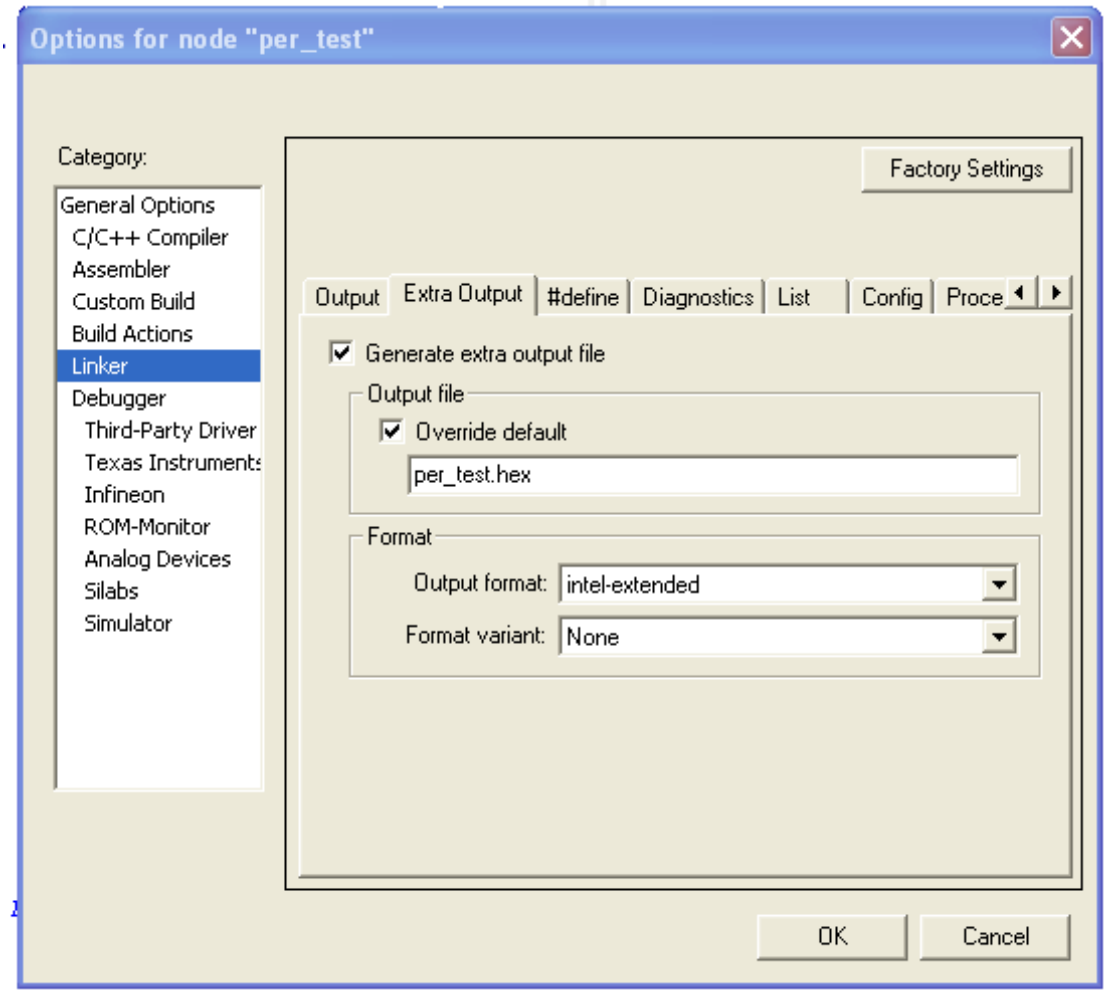


Figure 19: Generate HEX file as primary output

To produce a hex file for banked code, please see the manual named “SWRU038 IAR User Manual” available from www.ti.com.

Note that if you only select the HEX file as output from the linker, you cannot debug this particular image. To have **both** the hex file **and** the debug file output, select “Debug information for C-SPY” and “Allow C-SPY-specific extra output file”. Then select the “Extra Output” tab.



Select “Generate extra output file” and specify name of the .hex file.

6 Installed hex files

After installation of the Flash Programmer, a few hex files have been installed. The typical location of these files will be:
C:\Program Files\Texas Instruments\SmartRF Tools\Firmware

For each device there will be a subfolder with the latest firmware version:

.\SmartRF04EB	\srf04eb_bootloader.hex	SmartRF04EB bootloader.
	\cc2430db_bootloader.hex	CC2430DB bootloader
	\fw0400.hex	SmartRF04EB application.
.\SmartRF05EB	\usb_bootloader_srf05eb.hex	SmartRF05EB bootloader.
	\cebal_fw_srf05eb.hex	SmartRF05EB application.
.\CC Debugger	\usb_bootloader_srf05dbg.hex	CC Debugger bootloader.
	\cebal_fw_srf05dbg.hex	CC Debugger application.

7 Document history

Revision	Date	Description/Changes
1.6	2010-12-09	Updated with the changes of the SmartRF Flash Programmer version 1.10.x.
1.5	2009-02-11	New version with support for CC2530 and the CC Debugger.
1.4	2007-12-27	New version with support for SmartRF05EB and MSP430
1.3	2007-09-19	Update of images and description of hex files added.
1.2	2006-05-16	Minor changes
1.1	2006-02-16	Changed layout
1.0	2005-12-21	Initial release

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Communications and Telecom	www.ti.com/communications
DSP	dsp.ti.com	Computers and Peripherals	www.ti.com/computers
Clocks and Timers	www.ti.com/clocks	Consumer Electronics	www.ti.com/consumer-apps
Interface	interface.ti.com	Energy	www.ti.com/energy
Logic	logic.ti.com	Industrial	www.ti.com/industrial
Power Mgmt	power.ti.com	Medical	www.ti.com/medical
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video and Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless-apps

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2010, Texas Instruments Incorporated